Renault SAS Nissan Peugeot Citroën Automobile	Reference : **0001-1** Version **1.0**	Publication date **2009/01/29** Page 1/8

Standard ECU reprogramming
Part 1 - General description

Comment:
General overview of programming procedure concept

AUTHORS(S)	
Name : Gilles Michard (Renault SAS) Cédric Meunier (Peugeot Citroën Automobile)	

1 Revision summary

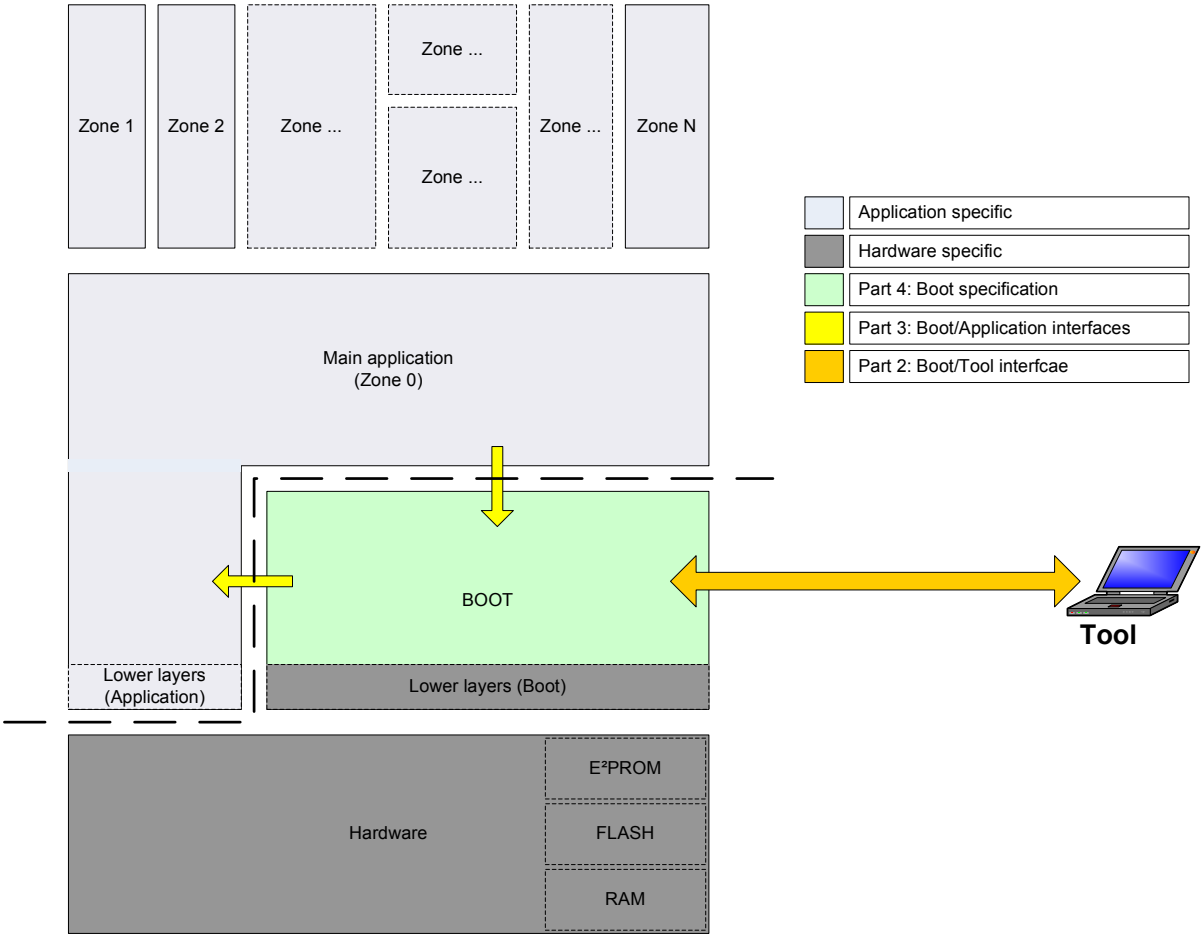
Revision	Date	Modified paragraphs and kind of modification
1.0	January 29th, 2009	First edition

2 Content

3 PURPOSE

This document is a part of Standard ECU reprogramming specification package. This package is divided into five parts, consistent to each others.

- Part 1: General description
- Part 2: Boot/tool interfaces description
- Part 3: Boot/Application interface description
- Part 4: Boot specification
- Part 5: Conformance test



Original documents will be freely available on "Internet Archive" website:
http://www.archive.org/

The Standard ECU reprogramming package contains specification on boot, how it interfere with application (application reprogramming and launch), with tool (how to reprogram an application, conformity test) and means to test the boot.

4 APPLICABLE DOCUMENTS

4.1 References documents

None

4.2 Norms and Procedures

Title	Ref.	Rev.
[1] Road vehicles — Unified diagnostic services (UDS) — Part 1: Specification and requirements	ISO 14229-1	2006
[2] Road vehicles — Diagnostics on Controller Area Networks (CAN) — Part 3:Implementation of unified diagnostic services (UDS on CAN)	ISO 15765-3	2004

5 TERMINOLOGY

5.1 Glossary

Segment	:	A segment is a in memory contiguous part of data.
Logical Block	:	A logical block is a block of consistent data that can be unitarily be reprogrammed (e.g. calibration or software module). A logical block is build of one or more segments.
Tool fingerprint	:	This is a data that represent the signature of the tool.
Digest	:	Result of a cryptographic hash function. It represents the calculated signature of a data set.

5.2 Abbreviations and acronyms

NRC	:	Negative Response Code. See document [1]

6 Overview

This document describes a mechanism for handling software and data updates in an ECU in conformance with the standard UDS programming steps which are currently described in ISO 15765-3:2004 and will be transferred into the next version of ISO 14229-1.

It is using hash functions to allow a tool to identify memory content without the need to actually read it, which could be time consuming; the ability to read back programmed data is not built in this specification: a tool can only decide if it has to change the content of the memory or to leave it unchanged. By separately writing the value of the hash function for the desired content and downloading the actual bytes, it gives the ECU the capacity to check if the full operation is successfully completed.

Any reason for the boot-loader to not start the applicative software is reflected in Diagnostic Trouble Codes (DTC). DTCs have a status with a “non tested” state, to represent the condition when a test result could have changed, but has still not been monitored.

Refer to HIS document.
“Terms and definitions”

6.1 Documents ownership

Those documents are built from a partnership between Renault SAS and Peugeot Citroën Automobile SA. This partnership is open to other companies.

The result of this partnership is free for use.

6.2 Programming steps

The following description is under the assumption that the programming application has already decided that un update to the memory of the ECU is needed. It can be because the ECU is still not operational, i.e. not fully programmed, or because a change in software or data must be done. The application has already chosen which logical blocks to update, knows the corresponding new digests and has access to the data to be downloaded.

6.2.1 Enter programming session

6.2.1.1 From boot-loader

In this case, the boot-loader was not authorized to start the application code.

6.2.1.2 From running application

In this case, the application is running. It must take appropriate checks and actions before giving control to the boot-loader.

6.2.2 Read programming counters

If a logical block reports 0 remaining operations (counter value 126), and that logical block needs to be updated, the operation must be cancelled: even if it could be possible to update another logical block, the operation can not succeed. It is probably better to leave the ECU in the current state.
If a negative number of operations is reported (counter value 127), the ECU is already unusable (a read of the DTC statuses would show a DTC with a testFailed status).

The programming application can inform the user if there is only one remaining programming operation left (a failure would mean the need to order a new part).

6.2.3 Unlock ECU

This step is not yet standardized.

6.2.4 Write target digests

Tell the ECU what the final state should be. Write the target « digest » of the areas to be updated. This is performed using WriteDataByIdentifier UDS service.

6.2.5 Download data

Using standard UDS RequestDownload/TransferData/TransferExit services.

6.2.6 Read DTC

At this step, at least DTC related to the downloaded logical blocks are reported as "not tested".
Except for the case of a logical block consisting of one sector: the update operation is atomic, meaning the logical bock is updated by only one download. It is allowed in that case to compute the new digest on the fly and to update the DTC status when the positive response to TransferExit is done.

6.2.7 Force DTC tests

A specific DTC test order shall be followed. The first DTC to check are the ones with fault type "not programmed". Other DTC like dependencies related can be then tested.

At the end of this step, either there is a failed DTC or all are tested and not failed. In first case boot-loader is not authorized to start application. The second case indicates that programming operations have been done successfully.

6.2.8 Reset ECU

Reset is performed either:

- Using the ECUReset UDS service.
- Exit the programming session using DiagnosticSessionControl UDS service.
- Exit the programming session because of session timeout.
- Power off/on the ECU.